UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Lucas Pandolfo Perin

FORMULAS FOR *P*-TH ROOT COMPUTATIONS IN FINITE FIELDS OF CHARACTERISTIC *P* USING POLYNOMIAL BASIS

Florianópolis

2016

Lucas Pandolfo Perin

FORMULAS FOR *P*-TH ROOT COMPUTATIONS IN FINITE FIELDS OF CHARACTERISTIC *P* USING POLYNOMIAL BASIS

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Prof. Daniel Panario, Dr.
Coorientador

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Perin, Lucas
Formulas for pth root computations in finite fields of characteristic p using polynomial basis / Lucas Perin ; orientador, Ricardo Custódio ; coorientador, Daniel Panario. - Florianópolis, SC, 2016.
65 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, . Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Cryptografia. 3. Teoria de números. 4. Aritmética em corpos finitos. 5. Base Polinomial. I. Custódio, Ricardo. II. Panario, Daniel. III. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Ciência da Computação. IV. Titulo.

In memory of my everyday companion.
My great friend, Gus.

AGRADECIMENTOS

First I would like to thank my family for the support over these couple of years. I am very lucky to have such great people in my life.

Secondly, I would like to express my deepest gratitude to Professor Qiang Wang and my advisors Professor Ricardo Felipe Custódio and Professor Daniel Panario. Their contributions were fundamental for this work. Thank you for your guidance.

Thirdly, a special thanks to André Castoldi and Lucas Boppre for numerous philosophical talks and assistance with mathematical problems, and to all my colleagues at LabSEC.

Last but not least, I would like to acknowledge and thank the examination board for all the contributions that have improved this master's thesis.

All you have to decide is what to do with the time that is given you.

J.R.R. Tolkien - The Fellowship of the Ring

RESUMO

Motivado por algoritmos criptográficos de emparelhamento bilinear, a computação da raiz cúbica em corpos finitos de característica 3 já fora abordada na literatura. Adicionalmente, novos estudos sobre a computação da raiz p-ésima em corpos finitos de característica p, onde p é um número primo, têm surgido. Estas contribuições estão centradas na computação de raízes para corpos de característica fixa ou para polinômios irredutíveis com poucos termos não nulos.

Esta dissertação propõe novas famílias de polinômios irredutíveis em $\mathbb{F}_p$, com k termos não nulos onde $k \geq 2$ e $p \geq 3$, para a computação eficiente da raiz p-ésima em corpos finitos de característica p. Além disso, para o caso onde $p = 3$, são obtidas novas extensões onde a computação da raiz cúbica é eficiente e polinômios cujo desempenho é ligeiramente melhor em comparação aos resultados da literatura.

Palavras-chave: Criptografia, Teoria de Números, Aritmética em Corpos Finitos.

ABSTRACT

Efficient cube root computations in extensions fields of characteristic three have been studied, in part motivated by pairing cryptography implementations. Additionally, recent studies have emerged on the computation of p-th roots of finite fields of characteristic p, where p prime. These contributions have either considered a fixed characteristics for the extension field or irreducible polynomials with few nonzero terms. We provide new families of irreducible polynomials over $\mathbb{F}_p$, taking into account polynomials with $k \geq 2$ nonzero terms and $p \geq 3$. Moreover, for the particular case $p = 3$, we slightly improve some previous results and we provide new extensions where efficient cube root computations are possible.

Keywords: Cryptography, Number Theory, Finite Field Arithmetic.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND INITIALS

LIST OF SYMBOLS

CONTENTS

1 INTRODUCTION

Over the years, elliptic curves have been extensively studied for their efficiency when used for cryptographic implementations. Today, for example, most websites that support the Transport Layer Security suite (TLS) are slowly transitioning to elliptic curves protocols and encryption algorithms. The experiments from Huang et al. (2014) show an optimistic result where a large fraction of the websites that support TLS prioritizes elliptic curves over RSA. This assumption is supported by the fact that most modern browsers implement Elliptic Curve Cryptography (ECC) suites. Therefore, if the client presents ECC as an option, the servers will use it since it is more efficient than RSA. The recommended key length of encryption algorithms from the National Institute of Standards and Technology (NIST, from the United States of America) is presented in Table 1. Taking into account the advances in cryptanalysis and the Moore's law, RSA key length will grow significantly when compared to ECC over the next decades. NIST recommends that 15360 bit RSA keys should be used to encrypt data safely after the year of 2030 (not taking *quantum computers* into account).

Table 1 – NIST security comparison of key length in bits.

Period of use	Symmetric	ECC	RSA
2011-2030	128 bits	256 bits	3072 bits
2030-?	256 bits	512 bits	15360 bits

Available at: `www.keylength.com`

As it turns out, elliptic curves can be used for other applications in cryptography. Bilinear pairings of points defined over elliptic curves have been proposed as an attack on ECC (MENEZES et al., 1993). Furthermore, pairings can also be used for cryptographic algorithms with remarkable flexibility, for example, the *Identity-based Encryption* scheme (IBE). Shamir (1985) first introduced the idea of IBE where two parties could communicate securely without exchanging randomly generated keys. He proposed that a public key - used to encrypt a session key - could be derived from a trivially authenticated parameter, such as an e-mail address. A central and trusted authority is responsible for this operation with the use of a master secret. This master secret is composed of a public and private master key, so that the authority may

publish the public master key and grant third parties the capability of computing any recipient's public keys. This way, it would be possible to send encrypted and authenticated messages even if the recipient has not yet created his own private key. However, a fully functioning IBE cryptosystem was only proposed years later, by Boneh and Franklin (2001), with the use of bilinear pairing. In the advent of this, the demand for fast Pairing-based Cryptography (PBC) algorithms have increased over the last couple of decades.

The study of efficient PBC is generally based on two main topics. The efficiency of the pairing construction itself and the implementation taking the target architecture into account. The first is usually centered on improving existing algorithms. This is done by proposing more efficient ways to compute arithmetic operations or modifications that decrease the overall complexity of the computation. The second is focused on the implementation of the algorithms that take advantage of the targeted computer architecture or even designing specific hardware for pairing computations.

One enhancement, in particular, is related to the main objective of this thesis. A pairing construction, proposed for a specific set of elliptic curves, uses cube root operations to compute the bilinear pairing. Consequently, efficient cube root extraction have been studied (BARRETO, 2004; AHMADI; HANKERSON; MENEZES, 2007; AHMADI; RODRÍGUEZ-HENRÍQUEZ, 2010). In addition, despite of these works being centered on the idea of computing cube roots (that is, a p-th root where $p = 3$), there has been recent developments in the computation of p-th roots, where p is prime (HARASAWA; SUEYOSHI; KUDO, 2006; PANARIO; THOMSON, 2009). In this thesis, we generalize previous cube root methods in order to compute p-th roots for finite fields of characteristic p, using polynomial basis.

1.1 MOTIVATION

The computation of p-th roots of elements expressed as polynomials is motivated by its use in factorization algorithms (GATHEN; PANARIO, 2001). In addition, it can also be used for PBC, where cube roots have been studied with the purpose of enhancing the performance of pairing operations over $\mathbb{F}_3$. However, recent developments have demonstrated that it is possible to compute the discrete log in extension fields of small characteristic in *quasi*-polynomial time (BARBULESCU et al., 2014). This suggests that cryptosystems that use such

extension fields could have much lower level of security than anticipated. For example, the extension field $\mathbb{F}_{3^{6 \cdot 509}}$ have been considered for the implementation of 128 bit security level cryptosystems. However, it turns out that pairing based cryptosystems using this field have significantly lower security level (ADJ et al., 2014). In light of this, the study of efficient p-th root computations may help leverage implementations of PBC using higher characteristics. Lastly, the overall study of efficient finite field arithmetic is a very interesting topic given the many areas where this arithmetic can be applied.

1.2 OBJECTIVES

1.2.1 General Objectives

The main goal of this work is to provide new extensions and families of irreducible polynomials over $\mathbb{F}_p$, that result on the efficient p-th root computation in finite fields of characteristic p, where p is an odd prime number.

1.2.2 Specific Objectives

- Obtain general equation for p-th root computations, where p is an odd prime number, for polynomials of varied sized of nonzero coefficients.
- Obtain general lower and upper bounds of the Hamming weight of our proposal, to compare to related work.

1.3 SCIENTIFIC METHOD

The main scientific method of this thesis is the mathematical proof of our proposals. We present 4 proved theorems, one corollary and a couple of propositions to illustrate the potentiality of the main theorems. An other method we use to evaluate the soundness of this thesis, is *SageMath*[1] mathematics software system. After the literature review is well established, simulations in search for patterns in the related work is well suited for our main objective. In addition, by using

[1] Available at `http://www.sagemath.org`

SageMath, it is also possible to evaluate the behavior of our proposed families of polynomials against each other.

1.4 SCIENTIFIC CONTRIBUTION

This work provides two families of irreducible polynomials that yield efficient p-th root computation in finite fields of characteristic p. These families include (a) "friendly" polynomials such that the p-th root computation requires no reduction modulo the irreducible polynomial defining the field and (b) "equally spaced" polynomials that have Hamming weight 1. These polynomials described in Chapter 5 and are part of an accepted paper to appear in the *Electronics Letters* journal (PERIN et al., 2015).

1.5 LIMITATIONS

In Chapter 3 we give a brief overview of bilinear pairings with the purpose of introducing algorithms that depend on the cube root computation. While the related work presented in Chapter 4 and our proposal presented in Chapter 5 have been motivated by bilinear pairings, we do not give detailed mathematical background of pairings algorithms in this thesis. The implementation of such algorithms using our proposal is considered for future works, in Chapter 6.

1.6 ORGANIZATION

The remainder of this thesis is organized as follows. The objectives, scientific method and contributions are discussed in this chapter. We give a basic mathematical background in Chapter 2, to facilitate the understanding of the main proposal of the thesis. In Chapter 3, we give a short description of bilinear pairings with the sole purpose of showing the algorithm where cube roots were introduced. In Chapter 4, we outline the main contributions of the related works mentioned above, with theorems and examples. The main contribution of this thesis is given in Chapter 5, where both families of polynomials are formally proven followed by implementation remarks. To give the final considerations of our work, Chapter 6 is the conclusion, followed by the references.

2 MATHEMATICAL BACKGROUND

In this chapter we give an introduction to the basic algebraic structures used in this thesis. Most definitions are taken directly from the book *Finite Fields* from Lidl and Niederreiter (1997). The definitions are presented with their reference to the book and followed by a few examples in the field of cryptography.

2.1 GROUPS

In mathematics, the operations of addition and multiplication, over the integers for example, are well established. However, the concept of operations to arbitrary sets can be generalized. Let $\mathbb{S}$ be a set and $\mathbb{S} \times \mathbb{S}$ denote the set of all ordered pairs (s, t) with $s, t \in \mathbb{S}$. Then a *binary operation* on $\mathbb{S}$ is a mapping from $\mathbb{S} \times \mathbb{S}$ to $\mathbb{S}$. When the image of the operation is the domain itself, the *closure* property is satisfied.

Definition 2.1.1 (Lidl and Niederreiter (1997), Definition 1.1)**.** A *group* $(\mathbb{G}, *)$ is a set $\mathbb{G}$ together with a binary operation $*$ on $\mathbb{G}$ such that the following three properties hold:

1 $*$ is *associative*; that is, for any $a, b, c \in \mathbb{G}$,

$$a * (b * c) = (a * b) * c;$$

2 there is an *identity* (or *unity*) *element* e in $\mathbb{G}$ such that for all $a \in \mathbb{G}$,

$$a * e = e * a = a;$$

3 for each $a \in \mathbb{G}$, there exists an *inverse element* $a^{-1} \in \mathbb{G}$ such that

$$a * a^{-1} = a^{-1} * a = e.$$

If the group satisfies $a * b = b * a$ for all $a, b \in \mathbb{G}$, then the group is *abelian* or *commutative*.

A set closed under an operation is an *algebraic structure*. From previous definition, $\mathbb{G}$ is an algebraic structure. Furthermore when $\mathbb{G}$ is closed under the multiplication, $\mathbb{G}$ is a *multiplicative group*.

Definition 2.1.2 (Lidl and Niederreiter (1997), Definition 1.3)**.** A multiplicative group $\mathbb{G}$ is said to be *cyclic* if there is an element $a \in \mathbb{G}$ such

that for any $b \in \mathbb{G}$ there is some integer j with $b = a^j$. Such element a is called a *generator* of the cyclic group, and we write $\mathbb{G} = \langle a \rangle$.

From the definition, we observe that a cyclic group is always commutative. A cyclic group may have more than one element that is a generator of the group.

Definition 2.1.3 (Lidl and Niederreiter (1997), Definition 1.6)**.** A group is called *finite* if it contains finitely many elements. The number of elements in a finite group is called its *order*.

Finite groups are often constructed using equivalence classes with the modulo of a positive integer. For example, let p be a prime number, then the set $\{[0], [1], \ldots, [p-1]\}$ closed under the operation $*$ is the *group of integers modulo p* denoted by $\mathbb{Z}_p$. We observe that elements of $\mathbb{Z}_p$ denoted by $[\alpha]$ for $0 \leq \alpha \leq p-1$ are equivalence classes such that

$$\begin{aligned}
[0] &= \{\ldots, -2p, -p, 0, p, 2p, \ldots\}, \\
[1] &= \{\ldots, -2p+1, -p+1, 1, p+1, 2p+1, \ldots\}, \\
&\vdots = \vdots \\
[p-1] &= \{\ldots, -(p-1), -1, p-1, 2p-1, \ldots\}.
\end{aligned}$$

Table 2 provides an example of an *additive* finite group of prime order, where $\mathbb{G}$ is closed under the operation of addition.

Table 2 – Sum in a finite group of three elements

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

When there exists a hard problem associated with a group, this property can be used in cryptography to construct secure protocols.

Example 2.1.4. *Diffie-Hellman Key Exchange Protocol*

Let $\mathbb{G}$ be a group of prime order q where $g \in \mathbb{G}$ is a generator. This protocol suggests that two parties - Alice and Bob - that wish to exchange private information, generate a common key to encrypt data through a public communication channel. Alice generates $a \in \mathbb{Z}_p$ and sends to Bob the value $A = g^a$, using the public channel. Bob does the same thing and sends $B = g^b$ to Alice, where $b \in \mathbb{Z}_p$ is

generated by Bob. Both parties may now compute g^{ab}. The security of this protocol relies on the Decision Diffie-Hellman (DDH) assumption, where $\{g, A, B, g^{ab}\}$ is indistinguishable from $\{g, A, B, g^{random}\}$ in $\mathbb{G}$.

Following from Example 2.1.4, other difficult problems used in the Diffie-Hellman (DH) protocol are the discrete log problem and the Computational Diffie-Hellman (CDH) problem. The discrete log problem is a very common tool used to build modern cryptosystems. It states that given g and g^x, the value x is hard to compute. The CDH is a hard problem where given g, g^x and g^y, it is hard to compute g^{xy}. In the Diffie-Hellman protocol from previous example, Alice computes the value g^{xy} given that she generated the value x in the first place.

2.2 FINITE FIELDS

Finite fields, also called Galois fields, are algebraic structures that contain a finite number of elements together with a set of properties and two binary operations. In this section we define the structures needed to construct finite fields, which are used in the remainder of the thesis.

Definition 2.2.1 (Lidl and Niederreiter (1997), Definition 1.28)**.** A *ring* $(\mathbb{R}, +, *)$ is a set $\mathbb{R}$, together with two binary operations, denoted by $+$ and $*$, such that:

1 $\mathbb{R}$ is an abelian group with respect to $+$;

2 $*$ is associative; that is, $(a * b) * c = a * (b * c)$ for all $a, b, c \in \mathbb{R}$;

3 *distributive laws* hold; that is, for all $a, b, c \in \mathbb{R}$ we have $a*(b+c) = a * b + a * c$ and $(b + c) * a = b * a + c * a$.

With the definition of ring, we may now briefly distinguish the algebraic structures that are obtained by further restricting rings with additional constraints.

Definition 2.2.2 (Lidl and Niederreiter (1997), Definition 1.28)**.**

i A ring is a *ring with identity* if it has a multiplicative identity; that is, if there is an element e such that $a * e = e * a = a$ for all $a \in \mathbb{R}$;

ii A ring is *commutative* if $*$ is commutative;

iii A ring is an *integral domain* if it is a commutative ring with identity $e \neq 0$ which $ab = 0$ implies that $a = 0$ or $b = 0$;

iv A ring is a *division ring* (or *skew field*) if the nonzero elements of $\mathbb{R}$ form a group under $*$;

v A commutative division ring is a *field*.

A field is a set $\mathbb{F}$ closed under the operation of addition and multiplication, where $\mathbb{F}$ is an abelian group with respect to the addition operation where 0 is the identity. Additionally, the nonzero elements of $\mathbb{F}$ form an abelian group with respect to the multiplication having, identity element $e \neq 0$.

Definition 2.2.3 (Lidl and Niederreiter (1997) Definition 1.41)**.** For a prime p, let $\mathbb{F}_p$ be the set $\{0, 1, \ldots, p-1\}$ of integers and let $\varphi : \mathbb{Z}/(p) \rightarrow \mathbb{F}_p$ be the mapping defined by $\varphi([a]) = a$ for $a = 0, 1, \ldots, p-1$. Then $\mathbb{F}_p$, endowed with the field structure induced by φ, is a finite field, the Galois field of order p.

The order of a finite field is the number of elements in the field. In the notation $\mathbb{F}_p$, p is the characteristic of the field. From Definition 2.2.3, since the characteristic is a prime number, the order of the field is also p. However, although the characteristic of a field is always a prime number, the order may not be. This is the case for extension fields, described in the following section.

In cryptography, recalling from Example 2.1.4, the DH protocol was first proposed where $\mathbb{G} = \mathbb{F}_p^*$ and p is a prime number. In addition, sub-exponential algorithms that solve the discrete log over finite fields have long existed (ODLYZKO, 1984). Therefore, for security reasons, p must be thousands of bits long.

2.3 EXTENSION FIELDS

Let $\mathbb{F}$ be a field. Then $\mathbb{K}$ is a *subfield* of $\mathbb{F}$ if $\mathbb{K}$ is a subset of $\mathbb{F}$ closed under the operations of $\mathbb{F}$. Then $\mathbb{F}$ is an *extension* (*field*) of $\mathbb{K}$.

Definition 2.3.1 (Lidl and Niederreiter (1997) Definition 1.77)**.** A field containing no proper subfields is a *prime field*.

A finite field of order p where p is prime is a *prime field*. The notation $\mathbb{F}_p$ is used to represent prime fields. Furthermore, to distinguish extension fields, we use the notation $\mathbb{F}_q$ where $q = p^m$ and p is the

characteristic of the field. In remainder of this thesis we treat extension fields as extensions of prime fields using a polynomial basis. Other cases such as normal basis and shifted basis are not treated and should be considered only if explicitly mentioned (LIDL; NIEDERREITER, 1997, page 54)

Definition 2.3.2 (Lidl and Niederreiter (1997), page 19)**.** A polynomial Q over $\mathbb{F}_q$ is an expression of the form $Q(x) = \sum_{i=0}^{n} a_i x^i$, where n is a nonnegative integer, and $a_i \in \mathbb{F}_q$ for $i = 0, 1, \ldots, n$. A polynomial is monic if the coefficient of the highest power of x is 1. The ring formed by the polynomials over $\mathbb{F}_q$ with sum and product is the ring of polynomials over $\mathbb{F}_q$ denoted by $\mathbb{F}_q[x]$.

Definition 2.3.3 (Lidl and Niederreiter (1997), Definition 1.57)**.** A polynomial $Q \in \mathbb{F}_q[x]$ is an irreducible polynomial over $\mathbb{F}_q$ if Q has positive degree and $Q = gh$ with $g, h \in \mathbb{F}_q[x]$ implies that either g or h is a constant polynomial.

From Definition 2.3.3, the polynomial Q is irreducible if it does not allow a factorization by two other polynomials of positive degree and $1 < \deg(g) \leq deg(h) < \deg(Q)$.

Let $\mathbb{F}_{p^m}$ be an extension field of characteristic p and order p^m. Then the notation $C \in \mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(Q)$, where Q is an irreducible polynomial over $\mathbb{F}_p$ with degree m, describes an element represented as the polynomial $C = \sum_{i=0}^{n} a_i x^i$ in the extension field $\mathbb{F}_{p^m}$, where the coefficients $a_i \in \mathbb{F}_p$ and $n < m$, similar to Definition 2.2.3.

Table 3 – Sum in a finite field of characteristic four

$+$	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

In Table 3 and 4, an example of the operations of addition and multiplication of elements in $\mathbb{F}_4 \cong \mathbb{F}_2[x]/(Q)$ where $Q(x) = x^2 + x + 1$ is given. The multiplication in this case is a little more complex than traditional arithmetic. Since the elements are polynomials, the multiplication and exponentiation usually result in polynomials of higher degrees. Also, since the largest degree exponent in the resulting polynomial might be higher than the exponent of the irreducible polynomial defining the field, a division by the irreducible polynomial that defines the field may be required. See the following example from Table 4.

Table 4 – Multiplication in a finite field of characteristic four

$\cdot$	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

Example 2.3.4.

$$(x+1)(x+1) = x^2 + 2x + 1 = x^2 + 1 = x \pmod{x^2 + x + 1}.$$

The last operation in Example (2.3.4) - where the result of the multiplication is the residue of the division by the irreducible polynomial defining the field extension - is called in this thesis a *reduction*.

3 BILINEAR PAIRING

The *Weil* pairing was introduced by *André Weil* in 1946, followed by the *Tate-Lichenbaum* pairing a few decades later. It was in the early 90's that pairings were first used as a cryptanalysis tool to "attack" the Discrete-log problem in select groups (MENEZES et al., 1993). Joux (2000) showed that it is possible to use pairings for cryptography when he proposed a three party key exchange protocol from the *Weil* pairing. Thereafter, bilinear pairings have been used for several new cryptographic protocols. An example is the Identity Based Encryption scheme with its first fully functional version proposed using the Weil pairing (BONEH; FRANKLIN, 2001).

Some of the works related to efficient p-th root computation are motivated by the computation of the modified Tate pairing, proposed by Duursma and Lee (2003). In this chapter, we briefly describe bilinear pairings and present the *Duursma-Lee* algorithm, to highlight the cube root operation that is later generalized in this work.

3.1 PAIRING DEFINITION

A bilinear pairing is a map function from two elements of a group to a target group.

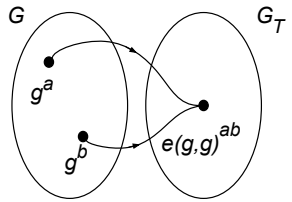

Figure 1 – Bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$

Definition 3.1.1. Let $\mathbb{G}$ be a group of points in an elliptic curve, $\mathbb{G}_T$ be an extension field and $g \in \mathbb{G}$ a generator of the group. A pairing is a non-degenerate bilinear map $\hat{e}$ where $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

- **Non-degenerate** if g is a generator of $\mathbb{G}$ then $\hat{e}(g, g)$ is a generator of $\mathbb{G}_T$.
- **Bilinear**: $\hat{e}(g, g)^{ab} = \hat{e}(g^a, g^b)$ for all $a, b \in \mathbb{G}$;

The bilinear property is used in cryptography where $\hat{e}(g^x, h^y) = \hat{e}(g,h)^{xy} = \hat{e}(g^y, h^x)$. In the following, we give an example of a signature scheme proposed by Boneh, Lynn and Shacham (2001).

Example 3.1.2. Boneh–Lynn–Shacham signature scheme:

Key generation. Choose x at random in $\mathbb{Z}_p$, then x is the private key and g^x is the public key.

Signature generation. Let $\mathbb{M}$ be a message where $\mathbb{M} \in \{0,1\}^*$ and $h = H(\mathbb{M})$ where H is a hash function. Then the signature s is computed as $s = h^x$.

Signature verification. Given the public key g^x, the message $\mathbb{M}$ and the signature s, verify if $\hat{e}(s, g) = \hat{e}(H(\mathbb{M}), g^x)$ holds.

If the signature is valid, then we have that $\hat{e}(h^x, g) = \hat{e}(h, g^x)$.

3.2 PAIRING COMPUTATION

Let $E(\mathbb{F}_{3^m})$ denote the points of the elliptic curve $E : y^2 = x^3 - x \pm d$ defined over $\mathbb{F}_{3^m}$. Let $P, Q \in E(\mathbb{F}_{3^m})$ where $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. The coordinates x and y are elements of the extension field and, in lines 3 and 8 of Algorithm 1, the cube root of these elements must be computed.

Algorithm 1: Duursma-Lee modified Tate pairing (m prime)

1 **Input:** $P, Q \in E(\mathbb{F}_{3^m})$;
2 **Output:** $\hat{e}(P,Q) \in \mathbb{F}_{3^{6m}}$;
3 $x_1 \leftarrow \sqrt[3]{x_1} - (v+1)b; \quad y_1 \leftarrow \lambda \sqrt[3]{y_1}$;
4 $y_2 \leftarrow -\lambda y_2$;
5 $t \leftarrow x_1 + x_2$;
6 $R \leftarrow \lambda(y_1 t - y_2\sigma - y_1\rho)(-t^2 + y_1 y_2 \sigma - t\rho - \rho^2)$;
7 **for** 1 *to* $\frac{m-1}{2}$ **do**
8 $\quad x_1 \leftarrow \sqrt[3]{x_1}; \quad y_1 \leftarrow \sqrt[3]{y_1}$;
9 $\quad x_2 \leftarrow x_2^3; \quad y_1 \leftarrow y_2^3$;
10 $\quad t \leftarrow x_1 + x_2; \quad u \leftarrow y_1 y_2$;
11 $\quad R \leftarrow R \cdot (-t^2 + u\sigma - t\rho - \rho^2)$
12 **end**
13 **Return** R^M;

For a more recent example of the implementation of this algorithm, we give yet another modified version of the Tate pairing. To further increase performance, Algorithm 1 can be enhanced to take advantage of multi-core processors (BEUCHAT et al., 2009). Lines 10 and 11 from Algorithm 1 may be optimized using *loop unrolling*. This technique is presented in lines 13 to 17 of Algorithm 2. By using this approach, it is possible to distribute the load of the operations in the multi-core pipeline. However, this comes with the cost of the storage of the computations in lines 10 and 11.

Algorithm 2: Duursma-Lee modified Tate pairing with loop unrolling (m prime)

1 **Input:** $P, Q \in E(\mathbb{F}_{3^m})$;
2 **Output:** $\hat{e}(P,Q) \in \mathbb{F}_{3^{6m}}$;
3 $x_1 \leftarrow \sqrt[3]{x_1} - (v+1)b; \quad y_1 \leftarrow \lambda\sqrt[3]{y_1}$;
4 $y_2 \leftarrow -\lambda y_2$;
5 $t \leftarrow x_1 + x_2$;
6 $R \leftarrow \lambda(y_1 t - y_2\sigma - y_1\rho)(-t^2 + y_1 y_2\sigma - t\rho - \rho^2)$;
7 $x_1[0] \leftarrow x_1; \quad y_1[0] \leftarrow y_1$;
8 $x_2[0] \leftarrow x_2; \quad y_2[0] \leftarrow y_2$;
9 **for** $j = 1$ *to* $\frac{m-1}{2}$ **do**
10 $\quad x_1[j] \leftarrow \sqrt[3]{x_1[j-1]}; \quad y_1[j] \leftarrow \sqrt[3]{y_1[j-1]}$;
11 $\quad x_2[j] \leftarrow x_2[j-1]^3; \quad y_1[j] \leftarrow y_2[j-1]^3$;
12 **end**
13 **for** $j = 1$ *to* $\frac{m-1}{4}$ **do**
14 $\quad t \leftarrow x_1[2j-1] + x_2[2j-1]; \quad u \leftarrow y_1[2k-1]y_2[2k-1]$;
15 $\quad t' \leftarrow x_1[2j] + x_2[2j]; \quad u' \leftarrow y_1[2k]y_2[2k]$;
16 $\quad S \leftarrow (-t^2 + u\sigma - t\rho - \rho^2)(-t'^2 + u'\sigma - t'\rho - \rho^2)$;
17 $\quad R \leftarrow RS$
18 **end**
19 **Return** R^M;

Both algorithms presented above use cube root computations repeatedly, in the main loop. Taking this into consideration, these algorithms are dependent on the performance of the cube root operation. The state of the art for the parallelization of the Tate pairing is available in "Parallelizing the Weil and Tate pairings" (ARANHA et al., 2011).

There exists a new trend on pairing algorithms that do not use cube roots. Recent approaches in efficient pairing computations are

defined over prime fields, rather then extension fields. The so-called *optimal* pairings have been reported to beat previous speed records of pairing computations. However, the outlining of these pairings along with their advantages and disadvantages is not related to the main scope of this thesis. For a survey and deeper understanding on the state of the art of bilinear pairings and optimal pairings, see "*The Realm of the Pairings*" (ARANHA et al., 2014).

4 LITERATURE REVIEW

Early works on efficient p-th root computations are centered on the idea of computing roots in finite fields with characteristic p for $p = 3$. Barreto (2004) achieved significant results by carefully selecting polynomials that result in efficient cube root computations. In his work, Barreto claims that the fastest algorithms known to compute the Tate pairing at that time was the Duursma-Lee algorithm (DUURSMA; LEE, 2003). The main loop of the algorithm uses cubing and cube root operations; this motivates his studies. In the following years other works on cube roots were published exploring polynomials with few nonzero coefficients over $\mathbb{F}_3$ (AHMADI; HANKERSON; MENEZES, 2007; AHMADI; RODRÍGUEZ-HENRÍQUEZ, 2010).

4.1 CUBE ROOTS

The overall idea of the cube root computation, or the *Folklore* algorithm using Barreto's notation, is as follows. Let $C \in \mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(Q)$, where Q is an irreducible polynomial of degree m, where $m = 3\omega + r$ for some positive integer ω and $0 \leq r \leq 2$. Then, $C = \sum_{n=0}^{m-1} c_n x^n = \sum_{n=0}^{3\omega+r-1} c_n x^n$ and

$$C = \sum_{n=0}^{\omega-1+\lceil \frac{r}{2} \rceil} c_{3n} x^{3n} + x \sum_{n=0}^{\omega-1+\lfloor \frac{r}{2} \rfloor} c_{3n+1} x^{3n} + x^2 \sum_{n=0}^{\omega-1} c_{3n+2} x^{3n},$$

where $c_n \in \mathbb{F}_3$ for $0 \leq n \leq m-1$. Then

$$C^{1/3} = \sum_{n=0}^{\omega-1+\lceil \frac{r}{2} \rceil} c_{3n} x^n + x^{1/3} \sum_{n=0}^{\omega-1+\lfloor \frac{r}{2} \rfloor} c_{3n+1} x^n + x^{2/3} \sum_{n=0}^{\omega-1} c_{3n+2} x^n. \quad (4.1)$$

We observe that the Folklore algorithm requires the computation of two multiplications by two fixed polynomials, $x^{1/3}$ and $x^{2/3}$. As mentioned before, multiplication of polynomials in finite fields may require reduction modulo Q. The multiplications for cube roots and the multiplication of two regular polynomials share the same complexity of $O(m^2)$ using classical arithmetic. However, the cube root operation may cost fewer operations. The reason for this is because the other factor of the product has degree smaller than $m/3$. Furthermore, since

the polynomials used in this case are fixed for any $C \in \mathbb{F}_{3^m}$, there is room for improvement. This is usually done by carefully selecting the irreducible polynomial defining the field.

Definition 4.1.1. The *Hamming weight* (wt) of a polynomial is the number of nonzero coefficients.

In this thesis we use repeatedly $wt(x^{1/3})$ and $wt(x^{2/3})$ for the polynomial representation of $x^{1/3}$ and $x^{2/3}$ in $\mathbb{F}_{3^m} = \mathbb{F}_3[x]/(Q)$, where Q is an irreducible polynomial over $\mathbb{F}_3$ of degree m. The Hamming weight is related to the performance of computing $C^{1/3}$ using Equation (4.1). It is also the main tool to compare the efficiency of families of polynomials in this thesis. Intuitively, by selecting irreducible polynomials with a small number of nonzero coefficients, the hamming weight should be small. Indeed, most studies so far have centered on this idea. In the following section, we demonstrate that this is true for some families of polynomials. However, it is possible to have polynomials with a large number of nonzero coefficients where $wt(x^{1/3}) = 1$. In adition, we prove this for higher characteristic p.

4.1.1 Trinomials

The trinomial representation of irreducible polynomials is advantageous for implementing modular reductions. Moreover, for m prime, if $Q(x) = x^m + ax^k + b$ and $m \equiv k \pmod 3$ then no modular reduction is needed for the computation of cube roots (BARRETO, 2004).

Theorem 4.1.2 (Barreto (2004))**.** *Let $Q(x) = x^m + ax^k + b$ be an irreducible polynomial over $\mathbb{F}_3$ where $m \equiv k \pmod 3$. Then*

$$wt(x^{1/3}) = \begin{cases} 3 & \text{if } m \equiv l \equiv 1 \pmod 3, \\ 2 & \text{if } m \equiv l \equiv 2 \pmod 3. \end{cases}$$

Example 4.1.3. *Case $m \equiv k \equiv 1 \pmod 3$.*
Let $m = 3\omega + 1$ and $k = 3\upsilon + 1$, then

$$b = -x^{3\omega+1} - ax^{3\upsilon+1}$$
$$bx^2 = -x^{3\omega+3} - ax^{3\upsilon+3}$$

Since $b \in \mathbb{F}_3$ and $b^2 = 1$, we have

$$\begin{aligned} x^{2/3} &= -bx^{\omega+1} - abx^{\upsilon+1} \\ x^{4/3} &= x^{2\omega+2} - ax^{\omega+\upsilon+2} + x^{2\upsilon+2} \\ x^{1/3} &= x^{2\omega+1} - ax^{\omega+\upsilon+1} + x^{2\upsilon+1}. \end{aligned}$$

Using Theorem 4.1.2, the complexity of Equation (4.1) is reduced to $O(m)$. To better see this improvement, let $C^{1/3} = C_1 + x^{1/3}C_2 + x^{2/3}C_3$ and $x^n C$ be denoted by $C^{<<n}$. Then from Example 4.1.3

$$\begin{aligned} C^{1/3} =& C_1 + C_2^{<<2\omega+1} - aC_2^{<<\omega+\upsilon+1} + C_2^{<<2\upsilon+1} \\ & - bC_3^{<<\omega+1} - abC_3^{<<\upsilon+1}. \end{aligned}$$

Example 4.1.4. *Case $m \equiv k \equiv 2 \pmod 3$.*
Let $m = 3\omega + 2$ and $k = 3\upsilon + 2$, then

$$\begin{aligned} x^{1/3} &= -bx^{\omega+1} - abx^{\upsilon+1} \\ x^{2/3} &= x^{2\omega+2} - ax^{\omega+\upsilon+2} + x^{2\upsilon+2}. \end{aligned}$$

If we consider $wt(C^{1/3}) = wt(x^{1/3}) + wt(x^{2/3})$ the total weight of $C^{1/3}$, then both cases of the theorem have the same total weight. From Example 4.1.4, observe that

$$\begin{aligned} C^{1/3} =& C_1 - bC_2^{<<\omega+1} - aC_2^{<<\upsilon+1} \\ & + C_3^{<<2\omega+2} - aC_3^{<<\omega+\upsilon+2} + C_3^{<<2\upsilon+2}. \end{aligned}$$

Therefore, from the previous examples, a notation to better evaluate the performance of the cube root operations - when using the Hamming weight as a tool - is the overall weight. Hence, in Theorem 4.1.2, we note that $wt(C^{1/3}) = 5$.

Further improvements were studied for trinomials, for example, where $m \equiv 0 \pmod 3$ and $k \equiv 1 \pmod 3$. Ahmadi, Hankerson and Menezes (2007) prove that, in this case, the Hamming weight of $x^{1/3}$ may be as low as 1 for specific polynomials. In the same work a full description of trinomials for efficient cube root computations is given. However, choosing polynomials that yield weight 1 may come with the cost of reduction modulo Q in Equation (4.1).

4.1.2 Tetranomials

Let $Q(x) = x^m + ax^k + bx^l + c$ be an irreducible polynomial over $\mathbb{F}_3$ where $m \equiv k \equiv l \equiv r \pmod 3$ with $r \in \{1, 2\}$. Similar to the trinomials presented in the previous section, these polynomials require no reduction modulo Q in (4.1).

Theorem 4.1.5 (Ahmadi and Rodríguez-Henríquez (2010))**.**
Let $Q(x) = x^m + ax^k + bx^l + c$ be an irreducible polynomial over $\mathbb{F}_3$ where $m \equiv k \equiv l \equiv r \pmod 3$ with $r \in \{1, 2\}$, then $wt(C^{1/3}) = 9$.

Example 4.1.6. *Case $r = 1$*
Let $m = 3\omega + 1$, $k = 3\upsilon + 1$ and $l = 3\nu + 1$. We have that

$$\begin{aligned} x^{2/3} =& - cx^{\omega+1} - acx^{\upsilon+1} - bcx^{\nu+1} \\ x^{1/3} =& x^{2\omega+1} - ax^{\omega+\upsilon+1} - bx^{\omega+nu+1} \\ & + x^{2\upsilon+1} + x^{2\nu+1} - abx^{\upsilon+nu+1}. \end{aligned}$$

Ahmadi and Rodríguez-Henríquez (2010), using the results of Theorem 4.1.5, denotes trinomials and tetranomials where $m \equiv k \equiv l \pmod 3$ holds as cube root friendly polynomials. In the next chapter we use the same shape of the exponents of the polynomials to generalize friendly polynomials irreducible over $\mathbb{F}_p$.

4.1.3 Pentanomials

In previous sections we show how cube root friendly polynomials have their weight affected when the number of nonzero coefficients is increased. Ahmadi and Rodríguez-Henríquez (2010) describe a family of polynomials that have more nonzero coefficients than the ones described before but lower Hamming weights.

Theorem 4.1.7 (Ahmadi and Rodríguez-Henríquez (2010))**.**
Let $Q(x) = x^{4d} + ax^{3d} + x^{2d} + cx^d + ac$ be an irreducible polynomial over $\mathbb{F}_3$ where d is a positive integer and $3 \nmid d$. Then, if $a = c$, $wt(x^{1/3}) = 1$ and $wt(x^{2/3}) = 1$.

Example 4.1.8. *Case $t \equiv 1 \pmod 3$*

$$
\begin{aligned}
x^{4d} &= -ax^{3d} - x^{2d} - cx^d - ac \\
x^{5d} &= -ax^{4d} - x^{3d} - cx^{2d} - acx^d \\
&= -a(-ax^{3d} - x^{2d} - cx^d - ac) - x^{3d} - cx^{2d} - acx^d \\
&= x^{3d} + ax^{2d} + acx^d + c - x^{3d} - cx^{2d} - acx^d \\
x^{5d+1} &= (a-c)x^{2d+1} + cx \\
cx &= -(a-c)x^{2d+1} + x^{5d+1} \\
x &= (1-ac)x^{2d+1} + cx^{5d+1} \\
x^{1/3} &= (1-ac)x^{\frac{2d+1}{3}} + cx^{\frac{5d+1}{3}} \\
x^{2/3} &= \left((1-ac)x^{\frac{2d+1}{3}} + cx^{\frac{5d+1}{3}}\right)^2 \\
x^{2/3} &= (ac-1)x^{\frac{4d+2}{3}} + (a-c)x^{\frac{7d+2}{3}} + x^{\frac{10d+2}{3}}
\end{aligned}
$$

The polynomials described in Theorem 4.1.7 are *equally spaced polynomials*, because of their construction where consecutive nonzero coefficients are spaced by d. These polynomials have been studied as an alternative to compute cube roots in $\mathbb{F}_{3^m}$ when friendly polynomials do not exist.

4.2 P-TH ROOTS

There exists only a few irreducible binomials in $\mathbb{F}_2$ and $\mathbb{F}_3$. For this reason, only polynomials with three or more nonzero coefficients have been considered in the previous section. Alternatively, binomials can be used in higher characteristics and result in weights equal to 1 for p-th root computations. As it turns out, by using Theorem 4.2.1, it is possible to prove the existence of infinite extensions of irreducible binomials over $\mathbb{F}_{p^m}$ with odd characteristic $p \geq 5$ and $p^m > 50$.

Theorem 4.2.1 (Panario and Thomson (2009))**.** *Let $\mathbb{F}_{p^m}$ be a finite field of odd characteristic p, $p \geq 5$. There exists an irreducible binomial over $\mathbb{F}_p$ of degree m, $m \not\equiv 0 \pmod 4$, if and only if every prime factor of m is also a prime factor of $p^m - 1$. For $m \equiv 0 \pmod 4$ then there exists an irreducible binomial over $\mathbb{F}_p$ of degree m if and only if $p^m \equiv 1 \pmod 4$ and every prime factor of m is also a prime factor of $p^m - 1$.*

First we demonstrate the case where $p = 5$. From the previous theorem, the irreducible binomials over $\mathbb{F}_5$ have degree $m = 2^k$. Let

$C \in \mathbb{F}_{5^m} \cong \mathbb{F}_5[x]/(Q)$, where Q is an irreducible polynomial with degree $m = 5\omega + r$, with $0 \leq r \leq 4$. Then for $m \equiv 4 \pmod 5$ (that is, $r = 4$) and with $0 \leq i \leq m-1$

$$\begin{aligned} C^{1/5} =& \left(\sum_{i \equiv 0 (\mathrm{mod} 5)} c_i x^{\frac{i}{5}} \right) + x^{1/5} \left(\sum_{i \equiv 1 (\mathrm{mod} 5)} c_i x^{\frac{i-1}{5}} \right) + \\ & x^{2/5} \left(\sum_{i \equiv 2 (\mathrm{mod} 5)} c_i x^{\frac{i-2}{5}} \right) + x^{3/5} \left(\sum_{i \equiv 3 (\mathrm{mod} 5)} c_i x^{\frac{i-3}{5}} \right) + \\ & x^{4/5} \left(\sum_{i \equiv 4 (\mathrm{mod} 5)} c_i x^{\frac{i-4}{5}} \right). \end{aligned} \tag{4.2}$$

If $Q(x) = x^m - b$ then $x^{1/5} = (-b)^e x^{e\omega + (er+1)/5}$ where e is the smallest positive integer such that $er \equiv -1 \pmod 5$. Values of e and r are given in the following table.

Table 5 – Values of e and r for p-th root computations using binomials

e	r	$(er+1)/5$
1	4	1
2	2	1
3	3	2
4	1	1

By taking powers of $x^{1/5}$ the values of $x^{2/5}$ to $x^{4/5}$ can be determined. Let $\gamma = e\omega + (er+1)/5$ then, using Barreto's notation, the computation of fith-roots using Equation (4.2) is the following:

$$\begin{aligned} C^{1/5} =& C_0 + (-b)^e C_1^{>>\gamma} + (-b)^{2e} C_2^{>>2\gamma} \\ & + (-b)^{3e} C_3^{>>3\gamma} + (-b)^{4e} C_4^{>>4\gamma}. \end{aligned}$$

For $m \equiv 1, 2, 3 \pmod 5$, the computation of $C^{1/5}$ is similar with exception of a few changes on the range of the summations. Equation (4.2) is a variation of the Folklore algorithm for characteristic 5 and Panario and Thomson (2009) give a general formula for this p-th root variation.

Theorem 4.2.2 (Panario and Thomson (2009))**.** *Let q be a power of an odd prime p and let m be a positive integer such that there exists an irreducible binomial $Q(x) = x^m - b$ over $\mathbb{F}_q$, as given by Theorem 4.2.1.*

After a precomputation of $2(p-1)$ *elements in* $\mathbb{F}_q$*, the* p*-th root of an element* $C \in \mathbb{F}_{q^m}$ *requires* $p-1$ *scalar multiplications of elements in* $\mathbb{F}_{q^m}$ *by elements in* $\mathbb{F}_q$*. In addition, the computation requires at most* $(p-1)\lceil m/p \rceil$ *additions in* $\mathbb{F}_{q^m}$*.*

Let $\gamma = e\omega + (er+1)/p$, then $x^{1/p} = b^{-e}x^{\gamma}$. Additionally, the values of $x^{2/p}$ to $x^{p-1/p}$ are obtained by taking powers of $x^{1/p}$.

$$C^{1/p} = C_0 + b^{-e}C_1^{>>\gamma} + \cdots + b^{-(p-1)}C_{p-1}^{>>(p-1)\gamma} \pmod{Q}. \quad (4.3)$$

The general p-th root algorithm for irreducible binomials is given in Equation 4.3. It is noticed that $\gamma < m$, however, since the following constants are obtained by taking powers of the previous ones, we observe that $(p-1)\gamma > m$. Therefore, this method for p-th root computations force us to carry out a reduction modulo Q.

4.3 SHIFTED POLYNOMIAL BASIS

Previous sections consider the cube root and p-th root problem in finite fields using polynomial basis. While the remainder of this thesis uses the same approach, in this section we give an example of cube roots in $\mathbb{F}_{3^m}$ using a shifted polynomial basis.

Definition 4.3.1. Let s be an integer and $Q = \{x^i | 0 \leq i \leq m-1\}$ be a polynomial basis over $\mathbb{F}_{3^m}$. Then $x^{-s}Q = \{x^{i-s} | 0 \leq i \leq m-1\}$ is a *shifted polynomial basis.*

From definition 4.3.1, we observe that the polynomial basis is a special case of the shifted polynomial basis, where $s = 0$. Shifted polynomial basis can be used to reduce the Hamming weights of $x^{1/3}$ and $x^{2/3}$ in Section 4.1. Additionally, for some cases, it is possible to determine a value for s such that no reduction is required (CHO; CHANG; HONG, 2014). Let $s = 3t + \alpha$ where t is a positive integer, $\alpha \in \{0, 1, 2\}$ and

$$\delta_j(j = 1, 2) = \begin{cases} 0 & \text{if } [\alpha + j] < 3, \\ 1 & \text{if } [\alpha + j] \geq 3. \end{cases}$$

Similarly to Equation (4.1), and letting $m \equiv 0 \pmod 3$ we have

$$C = x^{-(3t+\alpha)}\left(\sum_{n=0}^{\omega-1} c_{3n}x^{3n} + x\sum_{n=0}^{\omega-1} c_{3n+1}x^{3n} + x^2\sum_{n=0}^{\omega-1} c_{3n+2}x^{3n}\right),$$

where $c_n \in \mathbb{F}_3$ for $0 \leq n \leq m-1$, and

$$\begin{aligned} C &= \sum_{n=0}^{\omega-1} c_{3n+\alpha} x^{3n+\alpha-(3t\alpha)} + x \sum_{n=-\delta_1}^{\omega-1-\delta_1} c_{3n+\alpha+1} x^{3n+\alpha-(3t\alpha)} \\ &\quad + x \sum_{n=-\delta_2}^{\omega-1-\delta_2} c_{3n+\alpha+2} x^{3n+\alpha-(3t\alpha)} \\ C^{1/3} &= \sum_{n=0}^{\omega-1} c_{3n+\alpha} x^{n-t} + x^{1/3} \sum_{n=-\delta_1}^{\omega-1-\delta_1} c_{3n+\alpha+1} x^{n-t} \\ &\quad + x^{2/3} \sum_{n=-\delta_2}^{\omega-1-\delta_2} c_{3n+\alpha+2} x^{n-t} \end{aligned} \tag{4.4}$$

Theorem 4.3.2 (Cho, Chang and Hong (2014), Theorem 6)**.** *Let $Q(x) = x^m + ax^k + b$ be an irreducible polynomial over $\mathbb{F}_3$ where $m \equiv 0 \pmod 3$ and $k \equiv 1 \pmod 3$. Then $wt(C^{1/3}) = 5$.*

The proof of Theorem 4.3.2 is similar to previous examples of this chapter and the values of $x^{1/3}$ and $x^{2/3}$ are given below.

$$x^{1/3} = -ax^{\omega-\upsilon} - abx^{-\upsilon},$$

$$x^{2/3} = x^{2\omega-2\upsilon} + x^{-2\upsilon} - bx^{\omega-2\upsilon}.$$

We observe that negative exponents in $x^{1/3}$ and $x^{2/3}$ exist when $s \neq 0$. For the case where $s = 0$, further computations are required. Therefore, when $m \equiv 0 \pmod 3$ and $k \equiv 1 \pmod 3$, the Hamming weight of $C^{1/3}$ is lower when $s \neq 0$. Furthermore, in the following equation we impose the condition such that the degree of $C^{1/3}$ is smaller than m

$$\begin{cases} -2\upsilon - \delta_2 - t \geq -3t - \alpha, \\ 3\omega - 2\upsilon - i - \delta_2 - t \leq 3\omega - 1 - (3t + \alpha). \end{cases} \tag{4.5}$$

By replacing the constants $x^{1/3}$ and $x^{2/3}$ from Theorem 4.3.2 in Equation (4.4), the minimum degree of $C^{1/3}$ is $-2\upsilon - \delta_2 - t$ and maximum degree of $C^{1/3}$ is $3\omega - 2\upsilon - i - \delta_2 - t$. Then Equation (4.5) is satisfied when the ordered pair (t, α) is equal to $(\upsilon, 0)$ or $(\upsilon, 1)$. Hence, if $s = 3\upsilon$ or $s = 3\upsilon + 1$, no reduction modulo Q is required in Equation (4.4).

5 FORMULAS FOR *P*-TH ROOTS

In this chapter, we present new families of irreducible polynomials over $\mathbb{F}_p$ with more nonzero coefficients than the ones studied so far. We prove that these polynomials still have low weights for p-th root computations. These families include (a) "friendly" polynomials such that no reduction modulo the irreducible polynomial of degree m defining $\mathbb{F}_{p^m}$ is required, and (b) "equally-spaced polynomials" that have Hamming weight 1. When $p = 3$ we extend previous results by providing new families of irreducible polynomials and new extensions m with low cube root complexity. When $p = 5$, only binomials have been treated (PANARIO; THOMSON, 2009); we extend this to polynomials with more terms and for $p \geq 5$.

First we generalize the Folklore algorithm for p-th roots, so that the constants $x^{1/3}$ and $x^{2/3}$ in Equation (4.1) are represented as $x^{i/p}$ with $1 \leq i \leq p-1$. Let $C \in \mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(Q)$ where Q is an irreducible polynomial over $\mathbb{F}_p$ of degree m and $m = p\omega + r$ for some positive integer ω. Let $0 \leq r \leq p-1$, $G(i) = 1$ if $i < r$ and $G(i) = 0$ if $i \geq r$. Then,

$$C^{1/p} = \sum_{i=0}^{p-1} \left(x^{i/p} \sum_{n=0}^{\omega-1+G(i)} c_{pn+i} x^n \right). \tag{5.1}$$

We provide a description of each family of polynomial in the following sections. To further generalize our work, we use k-nomials in alternative to fixed trinomials or tetranomials. Upper and lower bounds for the Hamming weights of the p-th root friendly polynomials and a formal proof of existence for the equally spaced polynomials are provided.

5.1 *P*-TH ROOT FRIENDLY POLYNOMIALS

A polynomial with k nonzero coefficients is a k-nomial and it can be expressed as $Q(x) = a_0 + \sum_{n=1}^{k-1} a_n x^{\beta_n}$ for some β_n, $1 \leq n \leq k-1$.

Definition 5.1.1. Let $Q(x) = a_0 + \sum_{n=1}^{k-1} a_n x^{\beta_n}$ be an irreducible k-nomial over $\mathbb{F}_p$ of degree m, then Q is a *p-th root friendly polynomial* (or p-th root friendly k-nomial) if $\beta_1 \equiv \beta_2 \equiv \cdots \equiv \beta_{k-1} \equiv r \pmod{p}$ where $m = \beta_{k-1}$ and $0 \leq r \leq p-1$ holds.

In this section we show that the reduction modulo Q is not re-

quired for the computation of p-th roots when Q is a p-th root friendly k-nomial.

Let Q be a friendly k-nomial irreducible over $\mathbb{F}_p$. Then $Q(x) = a_0 + x^r f(x^p)$ where $f(x) = \sum_{n=1}^{k-1} a_n x^{(\beta_n - r)/p}$. Let u be the inverse of $p - r$ in $\mathbb{F}_p$ so that $u(p-r) \equiv 1 \pmod{p}$. We have that for each $1 \leq i \leq p-1$ there exists j_i and t_i with $1 \leq j_i, t_i \leq p-1$, such that $ui = pt_i + j_i$. In particular, when $i = p - r$, we have $j_{p-r} = 1$. Let $t_{p-r} = t'$; then we have $u(p-r) = pt' + 1$. Using standard computations in finite fields, we have $-a_0 x^{-r} = f(x^p)$ and so

$$\begin{aligned} -a_0 x^{p-r} &= x^p f(x^p) \\ (-a_0)^{1/p} x^{(p-r)/p} &= x f(x) \\ \left((-a_0)^{1/p} x^{(p-r)/p}\right)^{j_i} &= (x f(x))^{j_i}. \end{aligned}$$

Multiplying both sides by $\left((-a_0)^{1/p} x^{(p-r)/p}\right)^{pt_i}$ we get

$$\left((-a_0)^{1/p} x^{(p-r)/p}\right)^{pt_i + j_i} = (-a_0)^{t_i} x^{(p-r)t_i + j_i} f(x)^{j_i}. \tag{5.2}$$

Recalling that $ui = pt_i + j_i$, using Equation (5.2), and since for $1 \leq i \leq p-1$,

$$\left((-a_0)^{1/p} x^{(p-r)/p}\right)^{ui} = (-a_0)^{ui/p} x^{\frac{(pt'+1)i}{p}} = (-a_0)^{ui/p} x^{it'} x^{i/p},$$

we have

$$x^{i/p} = (-a_0)^{-j_i/p} x^{(p-r)t_i - it' + j_i} f(x)^{j_i}. \tag{5.3}$$

Using $u(p-r) = pt' + 1$, we obtain that $ur = up - pt' - 1 = p(u - t' - 1) + p - 1$. As $j_r = p - 1$ and $t_r = u - t' - 1$, we may now simplify the exponent $(p-r)t_i - it' + j_i$ of (5.3) obtaining

$$\begin{aligned} & pt_i - rt_i - it' + j_i \\ &= ui - rt_i - it' \\ &= (t_r + t' + 1)i - rt_i - it' \\ &= it_r - rt_i + i. \end{aligned}$$

Hence, for $1 \leq i \leq p-1$

$$x^{i/p} = (-a_0)^{-j_i/p} x^{it_r - rt_i + i} f(x)^{j_i}. \tag{5.4}$$

Theorem 5.1.2. *Let $C \in \mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(Q)$, where Q is a friendly k-nomial irreducible over $\mathbb{F}_p$ with degree m. Then there is no reduction*

modulo Q to compute the p-th roots with $x^{i/p}$, $1 \leq i \leq p-1$, in Equation (5.1).

Proof. We note that $-ur \equiv 1 \pmod{p}$, so $p|1+ur$. From Equation (5.4) we have

$$\begin{aligned} it_r - rt_i + i \leq r \text{ if and only if } & i(t_r+1) \leq r(t_i+1) \\ \text{if and only if } & i\left\lceil \frac{ur}{p} \right\rceil \leq r\left\lceil \frac{ui}{p} \right\rceil \\ \text{if and only if } & i\frac{ur+1}{p} \leq r\left\lceil \frac{ui}{p} \right\rceil \\ \text{if and only if } & i \leq pr\left\lceil \frac{ui}{p} \right\rceil - iur. \end{aligned}$$

Since $pr\left\lceil \frac{ui}{p} \right\rceil - iur \equiv i(-ur) \equiv i \pmod{p}$, we have $it_r - rt_i + i \leq r$. It is now possible to compute the degree of $x^{i/p}$ and determine if the computation of p-th roots require reductions modulo Q in Equation (5.1):

$$\begin{aligned} \deg(x^{i/p}) =& it_r - rt_i + i + \deg\big(f(x)\big)j_i \\ =& it_r - rt_i + i + \left(\frac{p\omega + r - r}{p}\right)j_i = it_r - rt_i + i + \omega j_i. \end{aligned}$$

If the highest degree in Equation (5.1) is greater or equal than m. Then

$$\begin{aligned} m \leq \deg\left(x^{i/p}\sum_{n=0}^{\omega-1+G(i)} c_{pn+i}x^n\right) \\ p\omega + r \leq \omega j_i + (it_r - rt_i + i) + \omega - 1 + G(i). \end{aligned}$$

We choose $i = r$; then we get the contradiction

$$\begin{aligned} p\omega + r \leq \omega(p-1) + r + w - 1 + G(r) \\ p\omega + r \leq p\omega + r - 1. \end{aligned}$$

□

The Hamming weight for p-th root friendly k-nomials can be computed directly from Equation (5.4) using the polynomial expansion of $f(x)^{j_i}$. Therefore, from the multinomial theorem, an upper bound

for $wt(x^{i/p})$ can be expressed as $\binom{j_i+k-2}{k-2}$.

Definition 5.1.3. Let Q be a p-th root friendly k-nomial. If $\beta_{k-1} - \beta_{k-2} = \cdots = \beta_2 - \beta_1 = p\alpha$ holds for $\alpha \in \mathbb{Z}_{>0}$, Q is an *improved p-th root friendly polynomial* (or k-nomial).

For the improved p-th root friendly polynomials, letting $\beta_1 = pz + r$ for some positive integer z, then $f(x) = \sum_{i=1}^{k-1} a_i x^{z+(i-1)\alpha}$ with $\omega = z + (k-2)\alpha$ in Equation (5.1). The result above shows that in this case, as expected, the computation of p-th roots in Equation (5.1) do not require a reduction modulo Q. Furthermore, the exponents of $f(x)$ are arranged in an arithmetic progression. Thus a lower upper bound can be achieved. To this end Theorem 5.1.4 and Corollary 5.1.6, on the field of additive combinatorics, are introduced.

Theorem 5.1.4. *Let $A = \{a_1, a_2, \ldots, a_k\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ be finite sets of positive integers arranged in an arithmetic progression with $n \geq k$. If the elements of A and B share the same difference d, the cardinality of the sumset of A and B is $|A| + |B| - 1$.*

When the elements of A and B share the same d, the assertion $a_w + b_x = a_y + b_z$ is true if and only if $w + x = y + z$, for all $a_w, a_y \in A$ and $b_x, b_z \in B$.

Example 5.1.5.

$$\begin{gathered} a_3 + b_5 = a_2 + b_6 \\ a_1 + 2d + b_1 + 4d = a_1 + d + b_1 + 5d \\ a_1 + b_1 + 6d = a_1 + b_1 + 6d \end{gathered}$$

Proof. Since A and B share the same d, the sumset of A and B may be represented as the following matrix. We note that the empty cells

are not used for the proof.

$$A+B = \left|\begin{array}{lllll}
a_1+b_1 & & & & \\
a_1+b_2 & a_2+b_1 & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
a_1+b_{k-1} & a_2+b_{k-2} & \cdots & a_{k-1}+b_1 & \\
a_1+b_k & a_2+b_{k-1} & \cdots & a_{k-1}+b_2 & a_k+b_1 \\
a_1+b_{k+1} & a_2+b_k & \cdots & a_{k-1}+b_3 & a_k+b_2 \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
a_1+b_n & a_2+b_{n-1} & \cdots & a_{k-1}+b_{n-k+2} & a_k+b_{n-k+1} \\
 & a_2+b_n & \cdots & a_{k-1}+b_{n-k+2} & a_k+b_{n-k+2} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & a_{k-1}+b_n & a_k+b_{n-1} \\
 & & & & a_k+b_n
\end{array}\right|$$

The elements contained in the same line of matrix $A+B$ satisfy the assertion above. Hence, all the distinct elements of the sumset are contained in the first and last columns. Let $m_{i,j}$ be an element of the matrix $A+B$. The result on sumsets entails the set $S = \{m_{1,1}, m_{2,1}, \ldots, m_{k,1}, m_{k+1,k}, m_{k+2,k}, \ldots, m_{k+n-2,k}, m_{k+n-1,k}\}$. Consequently $|S| = k+n-1 = |A|+|B|-1$. □

Corollary 5.1.6. *Let A be a finite set of positive integers arranged in an arithmetic progression. Let A^m be the sumset of A with itself repeated m times; then the cardinality of the sumset A^m is $m(|A|-1)+1$.*

Proof. By induction on m. Basis: using Theorem 5.1.4 for $m = 2$ we obtain

$$|A^2| = |A+A| = |A|+|A|-1 = 2(|A|-1)+1.$$

Let $A+A = B$. Since A and B share the same difference, for $m = 3$,

$$|A^3| = |A+A+A| = |A+B| = |A|+|B|-1 = 3(|A|-1)+1.$$

Inductive step: From the statement, when $m = k$ we have $|A^k| = k(|A|-1)+1$. Let $m = k+1$ then

$$\begin{aligned}
|A^{k+1}| = |A+A^k| &= |A|+|A^k|-1 \\
&= (|A|-1)+k(|A|-1)+1 \\
&= (k+1)(|A|-1)+1 \\
|A^m| &= m(|A|-1)+1.
\end{aligned}$$

□

Theorem 5.1.7. *Let $C \in \mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(Q)$, where Q is an improved friendly k-nomial irreducible over $\mathbb{F}_p$ of degree m. Then we have*

- $wt(x^{i/p}) = 1$ *for* $k = 2$;
- $wt(x^{i/p}) \geq 2$ *for* $k = 3$;
- $wt(x^{i/p}) \geq 3$ *for* $k \geq 4$;
- $wt(x^{i/p}) \leq j_i(k-2)+1$ *for* $k \geq 3$.

Proof. Recall that $f(x) = \sum_{n=1}^{k-1} a_n x^{(\beta_n - r)/p}$. When Q is a p-th root friendly irreducible binomial ($k = 2$), the expansion of $f(x)^{j_i}$ is trivial and always results in a monomial since $f(x)$ is also a monomial. Therefore, $wt(x^{i/p}) = 1$. The lower bounds when $k = 3$ and $k = 4$ are also trivial for $j_i = 1$. If $j_i > 1$ or if $k > 4$ then the lower bound is not trivially proven. we show this for $k = 4$ below

$$\begin{aligned} f(x) &= a_1x^1 + a_2x^2 + a_3x^3 \\ f(x)^2 &= a_1^2x^2 + a_1a_2x^3 + a_1a_3x^4 + a_1a_2x^3 + a_2^2x^4 + \ldots + a_3^2x^6 \\ &= a_1^2x^2 + 2a_1a_2x^3 + (a_1a_3 + a_2^2)x^4 + \ldots \end{aligned}$$

By taking powers of $f(x)$, the first and last two terms of the expansion are the only terms that have no additions in their coefficients. Therefore, these coefficients may never cancel since a_1, a_2 and a_3 are positive integers in $\mathbb{F}_p$. If the coefficients of Q are selected to cancel all the other coefficients of the expansion of $f(x)^{j_i}$, then the previous result holds for any $k \geq 4$.

The last item is the upper bound $j_i(k-2)+1$. The maximum number of coefficients in the expansion of $f(x)^{j_i}$ can be expressed as the cardinality of the sumset of the exponents of $f(x)$. Hence, from Corollary 5.1.6, $wt(x^{i/p}) = j_i(k-2)+1$. □

From the previous theorem, the overall weight of $C^{1/p}$ for the improved p-th root friendly polynomials, when $k \geq 4$, is

$$3(p-1) \leq wt(C^{1/p}) \leq (p-1)((k-2)p/2+1).$$

To show the potentiality of Theorem 5.1.7, we slightly improve the weights of $x^{1/3}$ and $x^{2/3}$ for the tetranomials described previously in Theorem 4.1.5. In the next proposition we provide the results for $r = 1$; the proof for $r = 2$ is similar.

Table 6 – Prime extensions where exists irreducible friendly tetranomials and improved friendly tetranomials over $\mathbb{F}_3$; $m < 510$.

m	Friendly/Improved	m	Friendly/Improved
11	-	251	$\star$
13	$\star$	263	$\star$
23	$\star$	277	$\star$
37	$\star$	311	-
47	$\star$	313	$\star$
59	$\star$	337	$\star$
61	$\star$	347	$\star$
71	$\star$	349	$\star$
73	$\star$	359	$\star$
83	$\star$	373	$\star$
97	$\star$	383	-
107	$\star$	397	-
109	$\star$	409	$\star$
131	$\star$	419	-
157	$\star$	421	$\star$
167	$\star$	431	$\star$
179	$\star$	433	$\star$
181	$\star$	443	-
191	$\star$	457	$\star$
193	$\star$	467	$\star$
227	$\star$	479	$\star$
229	$\star$	491	$\star$
239	$\star$	503	$\star$
241	$\star$		

($\star$) Both friendly and improved tetranomials exist; (-) Only Friendly tetranomials exist

Proposition 5.1.8. *Let $P(x) = x^m + ax^k + bx^l + c$ be an irreducible polynomial over $\mathbb{F}_3$ where $m \equiv k \equiv l \equiv r \pmod 3$ with $r = 1$. If $m - k = k - l = 3\alpha$ with $\alpha > 0$, then $wt(x^{2/3}) = 3$ and*

$$wt(x^{1/3}) = \begin{cases} 5 & \text{if } b = 2, \\ 4 & \text{if } b = 1. \end{cases}$$

Proof. Let $l = 3\omega + r$, $k = 3\omega + r + 3\alpha$ and $m = 3\omega + r + 6\alpha$ where $r = 1$. We have that $x^{1/3}$ can be calculated from $-c = x^m + ax^k + bx^l$. Indeed, this gives $-cx^2 = x^2(x^{3\omega+6\alpha+1} + ax^{3\omega+3\alpha+1} + bx^{3\omega+1}) = x^{3(\omega+2\alpha+1)} + ax^{3(\omega+\alpha+1)} + bx^{3(\omega+1)}$ and we conclude that

$$x^{2/3} = -cx^{\omega+2\alpha+1} - acx^{\omega+\alpha+1} - bcx^{\omega+1}.$$

Since a, b and c are nonzero in $\mathbb{F}_3$, and c satisfies $c^2 = 1$, we have $wt(x^{2/3}) = 3$. We get

$$\begin{aligned} x^{4/3} =& x^{2\omega+4\alpha+2} + x^{2\omega+2\alpha+2} + x^{2\omega+2} \\ & - ax^{2\omega+3\alpha+2} - bx^{2\omega+2\alpha+2} - abx^{2\omega+\alpha+2} \end{aligned}$$

implying that

$$\begin{aligned} x^{1/3} =& x^{2\omega+4\alpha+1} - ax^{2\omega+3\alpha+1} \\ & + (1-b)x^{2\omega+2\alpha+1} - abx^{2\omega+\alpha+1} + x^{2\omega+1}. \end{aligned}$$

□

We observe that the overall weight of $C^{1/3}$ is improved from $wt(C^{1/3}) = 9$ to $wt(C^{1/3}) = 7$. Furthermore, by further restricting the exponents of Q, it is expected that there exists fewer improved cube root friendly polynomials than there are regular cube root friendly polynomials. In contradiction to this, in Table 6 we show that - for m prime - most extensions where friendly polynomials exists, there also exists improved friendly polynomials [1].

5.2 P-TH ROOT EQUALLY SPACED POLYNOMIALS

Equally spaced polynomials can be used to achieve minimum overall weight when computing p-th roots using Equation 5.1.

[1] Source code available at `https://github.com/lucasperin/pth_root_finite_fields`

Definition 5.2.1. Let $Q(x) = x^{(k-1)d} + \sum_{n=0}^{k-2} a_n x^{nd}$ be an irreducible polynomial over $\mathbb{F}_p$ where d is a positive integer. If all coefficients a_n are nonzero, Q is an *equally spaced polynomial*.

In this section we show that if $Q(x) = x^{(k-1)d} + \sum_{n=0}^{k-2} \lambda^{k-n-1} x^{nd}$ is irreducible over $\mathbb{F}_p$ for $\lambda \in \mathbb{F}_p^*$, then $wt(x^{i/p}) = 1$ for $1 \leq i \leq p-1$ where $x^{i/p}$ are constants of Equation (5.1). By using Theorem 5.2.3, we also prove the existence of infinite equally spaced polynomials that are irreducible over $\mathbb{F}_p$.

Let $Q(x) = x^{(k-1)d} + \sum_{n=0}^{k-2} a_n x^{nd}$ be an irreducible polynomial over $\mathbb{F}_p$ where d is a positive integer. Then we have

$$\begin{aligned}
x^{(k-1)d} &= -\sum_{n=0}^{k-2} a_n x^{nd} \\
x^{(k-1)d} &= -a_{k-2} x^{(k-2)d} - \sum_{n=0}^{k-3} a_n x^{nd} \\
x^{kd} &= -a_{k-2} x^{(k-1)d} - \sum_{n=0}^{k-2} a_n x^{(n+1)d} \\
&= -a_{k-2}\Big(-\sum_{n=0}^{k-2} a_n x^{nd} \Big) - \sum_{n=0}^{k-3} a_n x^{(n+1)d} \\
&= a_{k-2}a_0 + \sum_{n=1}^{k-2} (a_{k-2}a_n - a_{n-1}) x^{nd} \\
a_{k-2}a_0 &= x^{kd} - \sum_{n=1}^{k-2} (a_{k-2}a_n - a_{n-1}) x^{nd}.
\end{aligned}$$

Then for some integer u where $1 \leq u \leq p-1$, multiplying both sides by x^u and letting $\gamma = (a_{k-2}a_0)^{-1}$

$$\begin{aligned}
a_{k-2}a_0 x^u &= x^{kd+u} - \sum_{n=1}^{k-2} (a_{k-2}a_n - a_{n-1}) x^{nd+u} \\
x^u &= \gamma x^{kd+u} - \gamma \sum_{n=1}^{k-2} (a_{k-2}a_n - a_{n-1}) x^{nd+u}.
\end{aligned} \tag{5.5}$$

Theorem 5.2.2. *Let $p \nmid k$ and $kd + u \equiv 0 \pmod{p}$ for some integer $1 \leq u \leq p-1$. Let Q be an irreducible equally spaced polynomial over $\mathbb{F}_p$ and let $a_n = \lambda^{k-n-1}$, $0 \leq n \leq k-2$, for some $\lambda \in \mathbb{F}_p$. Then*

$wt(x^{u/p}) = 1$, and this implies that $wt(x^{i/p}) = 1$ for $1 \leq i \leq p-1$.

Proof. From Equation (5.5), we impose the condition $a_{k-2}a_n = a_{n-1}$. Choose $n = k-2$, then

$$\begin{aligned} a_{k-2} &= \frac{a_{k-3}}{a_{k-2}} \\ a_{k-3} &= a_{k-2}^2 \\ a_{k-4} &= a_{k-2}^3 \end{aligned}$$

Successively we get

$$a_n = a_{k-2}^{k-n-1}.$$

By the choice of coefficients $a_n = \lambda^{k-n-1}$ with $1 \leq n \leq k-2$ and $\lambda \in \mathbb{F}_p$, we must have $a_{k-2}a_i = a_{i-1}$, and hence $x^u = \gamma x^{kd+u}$. Consequently, since $kd + u \equiv 0 \pmod{p}$, we always obtain $wt(x^{u/p}) = 1$ where

$$x^{u/p} = \gamma^{1/p} x^{\frac{kd+u}{p}}.$$

Recalling that u is constant and co-prime to p, then there exists integers t_i and j_i with $t_i \geq 0$ and $1 \leq j_i \leq p-1$ such that $ui = pt_i + j_i$ for $1 \leq i \leq p-1$. By taking powers of x^u, we have

$$\begin{aligned} x^{ui} &= \gamma^i x^{i(kd+u)} \\ x^{pt_i+j_i} &= \gamma^i x^{i(kd+u)} \\ x^{t_i+j_i/p} &= \gamma^{i/p} x^{i\frac{(kd+u)}{p}} \\ x^{j_i/p} &= \gamma^{i/p} x^{i\frac{(kd+u)}{p} - t_i}. \end{aligned} \tag{5.6}$$

We note that, by varying i in Equation (5.6), we obtain all values $j_i = \{1, 2, \ldots, p-1\}$, thus for $1 \leq i \leq p-1$ we have $wt(x^{i/p}) = 1$. □

Next, we show there always exist irreducible equally spaced k-nomials satisfying the requirements of Theorem 5.2.2. Let $Q(x) = x^{(k-1)} + \sum_{n=0}^{k-2} \lambda^{k-n-1} x^n$ and $y = x/\lambda$. Then, we can rewrite Q as

$$Q(x) = \lambda^{k-1} \sum_{n=0}^{k-1} \lambda^{-n} x^n = \lambda^{k-1} \sum_{n=0}^{k-1} y^n = \lambda^{k-1} R(y).$$

Then Q is irreducible over $\mathbb{F}_p$ if and only if R is irreducible over $\mathbb{F}_p$. Take any prime k such that order of p modulo k is $\phi(k)$. Then, R is an irreducible cyclotomic polynomial over $\mathbb{F}_p$ with $\deg(R) = k-1$ and $\text{ord}(R) = k$ (LIDL; NIEDERREITER, 1997, Theorem 2.42). Consequently, Q is also irreducible over $\mathbb{F}_p$. Furthermore, by Theorem 5.2.3 below, if k is an odd prime, $k \nmid (p^{k-1}-1)/k$ and d is any power of k, then $Q(x^d)$ is irreducible over $\mathbb{F}_p$.

Theorem 5.2.3 ((MENEZES et al., 2013) Theorem 3.9)**.** *Let $R \in \mathbb{F}_p[x]$ be an irreducible polynomial of degree $k-1$ and order k. Let d be a positive integer. Then $R(y^d)$ is irreducible over $\mathbb{F}_p$ if and only if $\gcd\left(d, \frac{p^{k-1}-1}{k}\right) = 1$, each prime factor of d divides k, and if $4 \mid d$, then $4 \mid p^{k-1}-1$.*

We show the potentiality of Theorem 5.2.2, combined with the irreducibility existence criterion just developed for k-nomials, to provide the exhaustive list of all irreducible equally spaced heptanomials (that is, polynomials with 7 nonzero terms and hence, $k = 7$) over $\mathbb{F}_3$. This gives new extensions of $\mathbb{F}_3$ with low Hamming weight. In the following, we denote a monic polynomial over $\mathbb{F}_3$ of the form $x^{6t} + a_5x^{5t} + a_4x^{4t} + a_3x^{3t} + a_2x^{2t} + a_1x^t + a_0$ by $a_5a_4a_3a_2a_1a_0$. The proof is given directly from Theorem 5.2.3.

Proposition 5.2.4. *An equally spaced heptanomial $Q \in \mathbb{F}_3[x]$ is irreducible if and only if it is one of the following:*

- *111111 or 212121 for $t = 7^i$ where $i \geq 0$;*
- *111122, 122122, 221112 or 222112 for $t = 2^i13^j$ where $i, j \geq 0$;*
- *121221, 122221, 221211 or 222211 for $t = 7^i13^j$ where $i, j \geq 0$;*
- *111112, 111222, 112222, 121212, 211212, 212122, 212212 or 222222 for $t = 2^i7^j13^h$ where $i, j, h \geq 0$.*

In Table 7, we give the Hamming weights of our heptanomials. In comparison to the equally spaced pentanomials in Section 4.1.3, these heptanomials have similar or better weight values. There are 21 new extensions for $m \leq 1024$ where equally spaced heptanomials exists but no polynomial with less number of nonzero terms exist. The extensions resulting on Hamming weights equal to 1 (6, 42, 294 for heptanomials) can be obtained from Theorem 5.2.2.

Table 7 – Hamming weights for equally spaced pentanomials and heptanomials over $\mathbb{F}_3$.

	Extensions	Coefficients $a_5a_4a_3a_2a_1a_0$	$t=1$ $x^{1/3}$	$t=1$ $x^{2/3}$	$t=2$ $x^{1/3}$	$t=2$ $x^{2/3}$
Penta	$4 \cdot 5^i$	0 1 1 1 1 1	1	1	1	1
		0 1 2 1 2 1				
	$4 \cdot 2^i 5^j$	0 1 1 1 2 2	2	3	3	2
		0 1 2 1 1 2				
Hepta	$6 \cdot 7^i$	1 1 1 1 1 1	1	1	1	1
		2 1 2 1 2 1				
	$6 \cdot 2^i 7^j 13^h$	1 1 1 1 1 2	2	2	2	2
		2 1 2 1 2 2				
		1 1 2 2 2 2	3	2	3	2
		2 1 1 2 1 2				

5.3 IMPLEMENTATION REMARKS

In this section we give a few remarks on the results of our implementations of p-th roots. Our goal is to contrast the effect of the reduction operation, required in the equally spaced polynomials and not in the friendly polynomials. The source code is available at the URL `https://github.com/lucasperin/pth_root_finite_fields` and it is written in *python*. The code can be executed on the Sage Environment which has its own implementation of polynomial arithmetic and finite fields.

Table 8 compares both families proposed in this thesis for the computation of p-th roots. The friendly polynomials, or friendly k-nomials, are computed for $k = 2$ to $k = 5$ and $k = 40$, so that the effect of the increasing Hamming weight can be measured. Furthermore, we give two computations of p-th roots using equally spaced polynomials, comparing the efficiency against each other and against the friendly polynomials. Timings for the computations are given in the last column of Table 8. These timings were measured in Sage v6.5, using the *timeit* package from python's standard library.

First, we point out that the results suggest that the friendly polynomials *always* have better performance than the equally spaced polynomials. However, this may be a misleading observation, since our implementation does not take advantage of low Hamming weights in the multiplications in Equation (5.1). Our benchmark application

Table 8 – Friendly and equally spaced polynomials benchmark ($\mathbb{F}_p$)

Extension	k-nomial	Family	μs
256	Binomial	Friendly	17.13
271	Trinomial	Friendly	17.41
271	Tetranomial	Friendly	17.45
271	Pentanomial	Friendly	17.45
294	Heptanomial	Eq. Spaced	21.08
272	17-nomial	Eq. Spaced	19.36
271	40-nomial	Friendly	17.47

Sage v6.5 - Intel x64 3.2GHz

runs over the multiplication of elements in the extension field. All of the constants $x^{i/p}$ of the binomial, the heptanomial and the 17-nomial have Hamming weight 1. However, the results demonstrate that the binomials are $\sim 13\%$ to $\sim 23\%$ faster than the equally spaced polynomials. In fact, the only difference in their computation, given that the input is the same element, is the degree of $x^{i/p}$. This indicates that the reduction entailed by the higher degrees of $x^{i/p}$ of the equally spaced polynomial family has significant impact in our implementation.

The friendly polynomials results are very similar, indicating that the Hamming weight have very little impact in our implementation. The performance gained on using binomials over 40-nomials of similar extension is $\sim 2\%$. Furthermore, if compared to *fewer*-nomials of the same extension, the impact is less than $\sim 0.4\%$. This result is expected since, as previously mentioned, our implementation does not take advantage of the low Hamming weights.

We conclude that the irreducible friendly polynomials have an advantage over other families, supported by the fact that they do not require reduction modulo the irreducible polynomial that defines the extension field. However, from our results, the performance of equally spaced polynomials compared to friendly k-nomials is not clear when $k > 2$. For a specific application and by using lower level programing language, the p-th roots can be computed by using *shifts* instead of regular "Sage multiplications". Therefore, the $\sim 13\%$ to $\sim 23\%$ difference from Table 8 could be much smaller.

6 FINAL CONSIDERATIONS

In this thesis we provide a general formula for computing p-th roots efficiently. Our method consists in choosing an appropriate irreducible polynomial to define the extension field. These polynomials are classified in two families, including (a) "friendly" polynomials such that the p-th root computation requires no reduction modulo the irreducible polynomial defining the field and (b) "equally spaced" polynomials that have Hamming weight 1.

We generalize Equation (4.1) and determine *formulae* to compute the values of $x^{i/p}$ for *friendly* and *equally spaced* polynomials. With this, we distinguish the p-th root *improved friendly polynomials* that have lower weights than the regular friendly polynomials. We give an upper and lower bound for the Hamming weights of $x^{i/p}$ of the p-th root friendly polynomials and for the improved case. When the friendly polynomial is a binomial, the Hamming weight is always 1.

Lastly - our implementations suggest that - whereas the equally spaced polynomials may have the lowest overall weight possible, the friendly polynomials family have better performance. This can be observed for the case $p \geq 5$, p prime, where friendly binomials exist and also have minimum overall Hamming weight. The p-th root computations with p-th root friendly binomials is around 13% faster than equally spaced polynomials of the similar degree. In light of this, we emphasize that the extensions where each family exists are not always the same, their comparison is merely for performance analysis. Equally spaced polynomials play an important role for p-th root computations when there are no irreducible friendly polynomial in a specific extension field.

For future work, we propose a deeper analysis on the performance impact of this thesis in real application algorithms. There is evidence that cube root friendly polynomials have been used to enhance pairing operations. However, the choice of irreducible polynomials for cryptosystems may also affect other operations used in the pairing algorithms. The reduction operation, for example, is also affected by the irreducible polynomials defining the extension field. We believe this could result in a promising trade-off study of p-th root friendly and *reduction-friendly* polynomials. Additionally, from a theoretical standpoint, a study of p-th root computations using *shifted polynomial basis* may lead to new families of polynomials and improvements for this operation.

REFERENCES

ADJ, G. et al. Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for discrete logarithm cryptography. In: *Pairing-Based Cryptography–Pairing 2013*. [S.l.]: Springer, 2014. p. 20–44.

AHMADI, O.; HANKERSON, D.; MENEZES, A. Formulas for cube roots in $\mathbb{F}_{3^m}$. *Discrete Applied Mathematics*, North-Holland, v. 155, n. 3, p. 260–270, 2007.

AHMADI, O.; RODRÍGUEZ-HENRÍQUEZ, F. Low complexity cubing and cube root computation over $\mathbb{F}_{3^m}$ in polynomial basis. *Computers, IEEE Transactions on*, IEEE, v. 59, n. 10, p. 1297–1308, 2010.

ARANHA, D. F. et al. The realm of the pairings. In: *Selected Areas in Cryptography–SAC 2013*. [S.l.]: Springer, 2014. p. 3–25.

ARANHA, D. F. et al. Parallelizing the weil and tate pairings. In: *Cryptography and Coding*. [S.l.]: Springer, 2011. p. 275–295.

BARBULESCU, R. et al. A heuristic *quasi*-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In: *Advances in Cryptology–Eurocrypt 2014*. [S.l.]: Springer, 2014. p. 1–16.

BARRETO, P. S. A note on efficient computation of cube roots in characteristic 3. *IACR Cryptology ePrint Archive*, v. 2004, p. 305, 2004.

BEUCHAT, J.-L. et al. Multi-core implementation of the tate pairing over supersingular elliptic curves. In: *Cryptology and Network Security*. [S.l.]: Springer, 2009. p. 413–432.

BONEH, D.; FRANKLIN, M. Identity-based encryption from the weil pairing. In: SPRINGER. *Advances in Cryptology—CRYPTO 2001*. [S.l.], 2001. p. 213–229.

BONEH, D.; LYNN, B.; SHACHAM, H. Short signatures from the weil pairing. In: *Advances in Cryptology—ASIACRYPT 2001*. [S.l.]: Springer, 2001. p. 514–532.

CHO, Y. I.; CHANG, N. S.; HONG, S. Formulas for cube roots in $\mathbb{F}_{3^m}$ using shifted polynomial basis. *Information Processing Letters*, Elsevier, v. 114, n. 6, p. 331–337, 2014.

DUURSMA, I.; LEE, H.-S. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. In: *Advances in cryptology-AsiaCrypt 2003*. [S.l.]: Springer, 2003. p. 111–123.

GATHEN, J. von zur; PANARIO, D. Factoring polynomials over finite fields: A survey. *Journal of Symbolic Computation*, Elsevier, v. 31, n. 1, p. 3–17, 2001.

GIRY, D. *Key Length Security Comparison*. 2015. `www.keylength.com/en`. Last accessed: 2016-01-18.

HARASAWA, R.; SUEYOSHI, Y.; KUDO, A. Ate pairing for $y^2 = x^5 - \alpha x$ in characteristic five. *IACR Cryptology ePrint Archive*, v. 2006, p. 114, 2006.

HUANG, L.-S. et al. An experimental study of tls forward secrecy deployments. *Internet Computing, IEEE*, IEEE, v. 18, n. 6, p. 43–51, 2014.

JOUX, A. A one round protocol for tripartite diffie–hellman. In: *Algorithmic number theory*. [S.l.]: Springer, 2000. p. 385–393.

LIDL, R.; NIEDERREITER, H. *Finite Fields*. [S.l.]: Cambridge university press, 1997.

MENEZES, A. J. et al. *Applications of Finite Fields*. [S.l.]: Springer Science & Business Media, 2013.

MENEZES, A. J. et al. Reducing elliptic curve logarithms to logarithms in a finite field. *Information Theory, IEEE Transactions on*, IEEE, v. 39, n. 5, p. 1639–1646, 1993.

ODLYZKO, A. M. Discrete logarithms in finite fields and their cryptographic significance. In: SPRINGER. *Advances in cryptology*. [S.l.], 1984. p. 224–314.

PANARIO, D.; THOMSON, D. Efficient pth root computations in finite fields of characteristic p. *Designs, codes and cryptography*, Springer, v. 50, n. 3, p. 351–358, 2009.

PERIN, L. P. et al. Formulas for pth root computations in finite fields of characteristic p. *Electronics Letters*, 2015.

SAGEMATH open-source mathematics software system. `http://www.sagemath.org/`. Last accessed: 2016-01-29.

SHAMIR, A. Identity-based cryptosystems and signature schemes. In: SPRINGER. *Advances in cryptology*. [S.l.], 1985. p. 47–53.